CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. (Original) A method for finding a low cost path from a source location to a target location through a traversable region partitioned into a plurality of clear tiles, each tile being an area outlined by a plurality of boundary segments, the method comprising:

searching for a path by propagating a cost function from at least one boundary segment in the traversable region to at least one other boundary segment;

recording the cost function for each boundary segment to which a cost is propagated;

backtracking from the target location to the source location using recorded cost functions for boundary segments after searching; and

providing a shortest path from the source location to the target location after backtracking.

2. (Original) The method of claim 1 wherein the searching for the path comprises:

propagating a source cost function from one of a plurality of boundary segments in the traversable region to at least one other of the plurality of boundary segments starting at the source location; and

repeating the propagating the source cost function until at least one source cost for the target location is found.

3. (Original) The method of claim 1 wherein the backtracking from the target location to the source location comprises:

selecting a previous boundary segment using a back pointer; and

selecting a path point on the selected segment by finding a point on the selected segment which provides a minimum value of the source cost function for the boundary segment.

4. (Original) The method of claim 3 further comprising repeating the selecting the previous boundary segment and the selecting the path point until the source is reached.

5. (Original) The method of claim 4 further comprising inserting the selected path point into a list of path points each time a path point is selected.

6. (Original) The method of claim 4 wherein the shortest path from the source location to the target location is derived at least in part from a list of the selected path points, each path point having been selected from a corresponding boundary segment.

7. (Original) The method of claim 3 wherein the selecting the path point comprises:

using a first path point selection criterion if the selected boundary segment is horizontal; and

using a second path point selection criterion if the selected boundary segment is vertical.

8. (Original) The method of claim 3 wherein the selecting the path point comprises:

analyzing a list of knot points on a previously recorded cost function corresponding to the selected boundary segment; and

selecting a knot point from the list of knot points which provides a minimum source cost for the selected boundary segment.

9. (Original) The method of claim 1 wherein

the plurality of boundary segments are prioritized based on a source cost for each boundary segment, and

the searching for the path continues until no boundary segments have a source cost lower than a recorded source cost of a boundary segment of the target tile.

10. (Currently amended) A method for finding a low cost path from a source location to a target location through a traversable region partitioned into a plurality of tiles, each tile being outlined by boundary segments, the method comprising the steps of:

selecting a boundary segment having a least cost of a plurality of boundary segments;

propagating a cost from the selected boundary segment to each boundary segment sharing a tile with the selected boundary segment;

recording the cost for each boundary segment ~~for use in finding the low cost path~~.

11. (Currently amended) ~~The method of claim 10~~ A method for finding a low cost path from a source location to a target location through a traversable region partitioned into a plurality of tiles, each tile being outlined by boundary segments, the method comprising the steps of:

selecting a boundary segment having a least cost of a plurality of boundary segments, wherein the first boundary segment selected is a boundary segment on a tile of the source, ~~the method further comprising:~~;

propagating a cost from the selected boundary segment to each boundary segment sharing a tile with the selected boundary segment;

recording the cost for each boundary segment;

repeating the steps of selecting the boundary segment, propagating the source cost and recording the cost until a cost has been recorded for a target boundary segment; and

backtracking from the target location to the source location using the recorded cost functions for each boundary segment.

12. (Currently amended) The method of claim [[10]] 11 wherein

the least cost is a path cost function providing a function for a path from the source location to the target location through points on the boundary segment;

the cost propagated from the selected boundary segment is a source cost function providing a function for a path from the source location to points on the selected boundary segment; and

the recorded cost is a source cost function.

13. (Currently amended) The method of claim [[10]] 11 wherein the step of selecting the boundary segment having the least path cost of the plurality of boundary segments comprises:

prioritizing boundary segments based on path costs of the boundary segments in a priority queue; and

popping a boundary segment from the priority queue which has the least path cost of the boundary segments in the priority queue.

14. (Currently amended) The method of claim [[10]] 11 further comprising:

assigning initial costs to boundary segments;

calculating a path cost for each boundary segment using an estimated target cost for each tile boundary segment and one of an initial source cost for each source tile boundary segment and a propagated source cost for each source tile boundary segment; and

continuously prioritizing the boundary segments based on path costs of the boundary segments.

15. (Original) The method of claim 14 further comprising:

repeating the steps of selecting the boundary segment, propagating the source cost and calculating the path cost until there are no further boundary segments to select.

16. (Original) The method of claim 14 further comprising repeating the steps of selecting the boundary segment, propagating the source cost and calculating the path cost until:

a target location boundary segment has been selected and a path cost has been calculated for the target location boundary segment; and

a next selected boundary segment has a path cost greater than the path cost for the target tile boundary segment.

17. (Original) The method of claim 14 wherein the assigning of initial costs comprises:

initializing source costs of source tile boundary segments to a first initial value; and

initializing costs of clear tile boundary segments and target tile boundary segments to a second initial value.

18. (Original) The method of claim 17 wherein

the first initial value is zero; and

the second initial value is an upper cost value.

19. (Original) The method of claim 13 wherein the estimated target cost is a weighted Manhattan distance between the boundary segment and the target tile.

20. (Original) The method of claim 13 wherein the target location is a target tile, and the calculating the path cost of each boundary segment comprises:

finding a convex hull containing the selected boundary segment and the target tile;

calculating a minimum cost function from the selected boundary segment to each target tile boundary segment not on the convex hull; and

assigning as a path cost a minimum of the calculated minimum cost functions for the target tile boundary segments not on the convex hull.

21. (Original) The method of claim 20 wherein the calculating the minimum cost function from the selected boundary segment to each target tile boundary segment not on the convex hull comprises:

using a perpendicular-specific cost propagation algorithm for each selected boundary segment that is perpendicular to each target tile boundary segment not on the convex hull; and

using a parallel-specific cost propagation algorithm for each selected boundary segment that is parallel to each target tile boundary segment not on the convex hull.

22. (Original) A computer program product encoded in at least one computer readable medium for finding a low cost path from a source location to a target location through a traversable region partitioned into a plurality of clear tiles, each tile being an area outlined by a plurality of boundary segments, the product comprising:

a first software module, executable by an information processing system, for searching for a path by propagating a cost function from at least one boundary segment in the traversable region to at least one other boundary segment;

a second software module, executable by an information processing system, for recording the cost function for each boundary segment to which a cost is propagated; and

a third software module, executable by an information processing system, for backtracking from the target location to the source location using recorded cost functions for boundary segments; and

a fourth software module, executable by an information processing system, for providing a shortest path from the source location to the target location after execution of the third software module.

23. (Original) The product of claim 22 wherein the first software module comprises:

at least one instruction for propagating a source cost function from one of a plurality of boundary segments in the traversable region to at least one other of the plurality of boundary segments starting at the source location; and

at least one instruction for repeatedly executing the instructions for propagating the source cost function until at least one source cost for the target location is found.

24. (Original) The product of claim 22 wherein the third software module comprises:

at least one instruction for selecting a previous boundary segment using a back pointer; and

at least one instruction for selecting a path point on the selected segment by finding a point on the selected segment which provides a minimum value of the source cost function for the boundary segment.

25. (Original) The product of claim 24 further comprising at least one instruction for repeating the selecting the previous boundary segment and the selecting the path point until the source is reached.

26. (Original) The product of claim 24 wherein the at least one instruction for selecting the path point comprises:

at least one instruction for using a first path point selection criterion if the selected boundary segment is horizontal; and

at least one instruction for using a second path point selection criterion if the selected boundary segment is vertical.

27. (Original) The product of claim 24 wherein the at least one instruction for selecting the path point comprises:

at least one instruction for analyzing a list of knot points on a previously recorded cost function corresponding to the selected boundary segment; and

at least one instruction for selecting a knot point from the list of knot points which provides a minimum source cost for the selected boundary segment.

28. (Currently amended) The product of claim 22 further comprising:

at least one instruction for prioritizing the plurality of boundary segments based on a source cost for each boundary segment; and

at least one instruction for the searching for the path ~~continues~~ until no boundary segments have a source cost lower than a recorded source cost of a boundary segment of the target tile.

29. (Original) The product of claim 22 wherein the at least one computer readable medium comprises at least one of a data storage medium and a data transmission medium, the data storage medium including at least one of the group consisting of a disk, tape, a compact disc and a digital video disc, and the data transmission medium including at least one of the group consisting of the Internet, a wireline or telecommunications network, and a wireless network.

30. (Original) The product of claim 22 embodied with an information processing system including a processor coupled to the at least one computer readable medium wherein the processor is configured to execute instructions encoded on the computer readable medium.